DL HW 3

https://github.com/sabbellasri/Deep-learning/tree/main/hw3

Task

Implement an Extractive Question and Answering system using a Bert model on the SQuAD dataset. The dataset consists of passages (contexts) accompanied by questions and their respective answers. The task requires feeding the context and the question as inputs to the model, which then predicts the start and end positions of the answer within the context.

Initially, utilize any base version of the Bert model from Hugging Face's library to process the SQuAD dataset. Evaluate the model's performance using the Word Error Rate (WER) metric. Subsequently, explore enhancements such as integrating a Linear Learning Rate Decay scheduler, optimizing the Document Stride parameter, refining Data Preprocessing techniques, and experimenting with alternative pretrained models to improve accuracy.

SQuAD Dataset:

This dataset, which is unaltered from its presentation in Homework 3 (HW3), comprises both training and testing JSON files. The training file features 37,111 question and answer pairs, each paired with its respective context. Meanwhile, the testing file includes 15,875 question and answer pairs.

Used Parameters:

Name :Model Optimizer

Value : bert-base-uncased AdamW

Learning rate: 2e-5

Weight Decay 2e-2

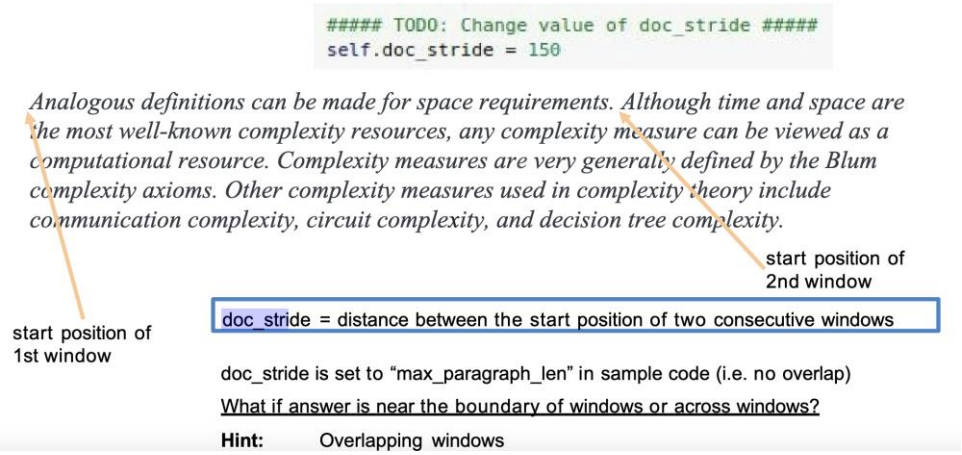
Dropout 0.1

MAX_LENGTH 512

Doc_stride 128

Doc Stride:

In the context of BERT-based model applications on lengthy texts, the "doc_stride" parameter plays a crucial role. It addresses the challenge of documents exceeding the model's maximum sequence length by dividing them into smaller, manageable segments, referred to as "strides." The doc_stride determines the overlap between these segments by specifying the token count by which to shift the sliding window across the document. This overlapping technique ensures that the model retains contextual coherence when analyzing various sections of the document



Scheduler:

The term "Scheduler" pertains to the mechanism of adjusting the learning rate during the BERT model training phase. This adjustment of the learning rate over time is critical for enhancing the training dynamics and ultimately the model's effectiveness. A prevalent approach involves initiating training with a relatively high learning rate to expedite early convergence. Subsequently, the learning rate is gradually reduced, facilitating more refined weight adjustments in the model. This strategy helps in precisely tuning the model's parameters without surpassing the optimal points in the loss function's landscape. Strategies such as linear decay of the learning rate, alongside initial warm-up periods, are widely adopted for this purpose.

Word Error Rate: Word Error Rate (WER) is predominantly a benchmark used within the realm of speech recognition to gauge the fidelity of transcribed text in comparison to a reference

script, by tallying the least number of insertions, deletions, and substitutions required. Applying WER directly to assess a BERT model's performance in question answering scenarios is atypical. Question answering evaluations concentrate on the model's capability to pinpoint the precise answer or its contextually accurate alternatives, rather than measuring the verbatim accuracy of transcribed speech.

This Python code is for training a question-answering model using BERT on a dataset containing contexts, questions, and answers. It first imports necessary libraries and defines paths for data and figure saving. The get_data function reads JSON files, extracts contexts, questions, and answers, and preprocesses them. Tokenization is done using a BERT tokenizer with truncation and padding. Start and end positions of answers are computed within the tokenized contexts. A custom PyTorch dataset class is defined, and data loaders are created. The BERT model is loaded, and a custom question-answering model is defined. Focal loss is used for training, and the AdamW optimizer is employed. Training and validation functions are defined to train and evaluate the model, respectively. The Word Error Rate metric is computed after each epoch for evaluation.

Base model:

```
/home/sabbels/.local/lib/python3.9/site-packages/transformers/optimization.py:457: FutureWarning: This implementation of
AdamW is deprecated and will be removed in a future version. Use the PyTorch implementation torch.optim.AdamW instead, or
set `no_deprecation_warning=True` to disable this warning
  warnings.warn(
Epoch - 0
Running Epoch : 100%|██████████| 2320/2320 [06:21<00:00,  6.08it/s]
Train Accuracy: 0.5263315886772912      Train Loss: 1.5555219033154948
Running Evaluation: 100%|██████████| 15875/15875 [03:22<00:00, 78.59it/s]
Epoch - 1
Running Epoch : 100%|██████████| 2320/2320 [06:22<00:00,  6.07it/s]
Train Accuracy: 0.6882446890504196      Train Loss: 0.812912741001567
Running Evaluation: 100%|██████████| 15875/15875 [03:21<00:00, 78.60it/s]
Epoch - 2
Running Epoch : 100%|██████████| 2320/2320 [06:22<00:00,  6.07it/s]
Train Accuracy: 0.7790871305712338      Train Loss: 0.5150295378222804
```

Base model with docstride and scheduler addition

```
/home/sabbels/.local/lib/python3.9/site-packages/transformers/optimization.py:457: FutureWarning: This implementation of
AdamW is deprecated and will be removed in a future version. Use the PyTorch implementation torch.optim.AdamW instead, o
set `no_deprecation_warning=True` to disable this warning
  warnings.warn(
Epoch - 0
Running Epoch : 100%|██████████| 2320/2320 [22:00<00:00,  1.76it/s]
Train Accuracy: 0.4130464901569588      Train Loss: 2.150769875763819
Running Evaluation: 100%|██████████| 15875/15875 [03:52<00:00, 68.25it/s]
Epoch - 1
Running Epoch : 100%|██████████| 2320/2320 [22:00<00:00,  1.76it/s]
Train Accuracy: 0.6144127155300873      Train Loss: 1.1390018417210928
Running Evaluation: 100%|██████████| 15875/15875 [03:52<00:00, 68.38it/s]
Epoch - 2
Running Epoch : 100%|██████████| 2320/2320 [22:00<00:00,  1.76it/s]
Train Accuracy: 0.7304475831574407      Train Loss: 0.690917913204637
Running Evaluation: 100%|██████████| 15875/15875 [03:51<00:00, 68.44it/s]
Word Error Rate-  [5.159622047244095, 3.9036220472440943, 4.063307086614174]
```

Base model with preprocessed data

```
/home/sabbels/.local/lib/python3.9/site-packages/transformers/optimization.py:457: FutureWarning: This implementation of
AdamW is deprecated and will be removed in a future version. Use the PyTorch implementation torch.optim.AdamW instead, o
set `no_deprecation_warning=True` to disable this warning
  warnings.warn(
Epoch - 0
Running Epoch : 100%|██████████| 2320/2320 [05:16<00:00,  7.33it/s]
Train Accuracy: 0.47056650247039467      Train Loss: 1.759330913578642
Running Evaluation: 100%|██████████| 15875/15875 [03:21<00:00, 78.73it/s]
Epoch - 1
Running Epoch : 100%|██████████| 2320/2320 [05:15<00:00,  7.35it/s]
Train Accuracy: 0.6705530326556542      Train Loss: 0.8639526987962168
Running Evaluation: 100%|██████████| 15875/15875 [03:21<00:00, 78.80it/s]
Epoch - 2
Running Epoch : 100%|██████████| 2320/2320 [05:15<00:00,  7.35it/s]
Train Accuracy: 0.7778748460627836      Train Loss: 0.5137672664533401
Running Evaluation: 100%|██████████| 15875/15875 [03:21<00:00, 78.63it/s]
Word Error Rate-  [4.190173228346457, 3.649259842519685, 5.234897637795275]
```

Pretrained Strong base model

```
/home/sabbels/.local/lib/python3.9/site-packages/transformers/optimization.py:457: FutureWarning: This implementation of
AdamW is deprecated and will be removed in a future version. Use the PyTorch implementation torch.optim.AdamW instead, or
set `no_deprecation_warning=True` to disable this warning
  warnings.warn(
Epoch - 0
Running Epoch : 100%|██████████| 2320/2320 [05:15<00:00,  7.35it/s]
Train Accuracy: 0.5261160714359119      Train Loss: 1.5631006753136372
Running Evaluation: 100%|██████████| 15875/15875 [02:48<00:00, 94.46it/s]
Epoch - 1
Running Epoch : 100%|██████████| 2320/2320 [05:15<00:00,  7.36it/s]
Train Accuracy: 0.7040294027277108      Train Loss: 0.769914671547454
Running Evaluation: 100%|██████████| 15875/15875 [02:48<00:00, 94.25it/s]
Epoch - 2
Running Epoch : 100%|██████████| 2320/2320 [05:13<00:00,  7.40it/s]
Train Accuracy: 0.8070062346242625      Train Loss: 0.4404591417140809
Running Evaluation: 100%|██████████| 15875/15875 [02:49<00:00, 93.80it/s]
Word Error Rate-  [4.580157480314961, 4.06992125984252, 3.6583307086614174]
```

Boss model

```
/home/sabbels/.local/lib/python3.9/site-packages/transformers/optimization.py:457: FutureWarning: This implementation of
AdamW is deprecated and will be removed in a future version. Use the PyTorch implementation torch.optim.AdamW instead, or
set `no_deprecation_warning=True` to disable this warning
  warnings.warn(
Epoch - 0
Running Epoch : 100%|██████████| 2320/2320 [06:21<00:00,  6.08it/s]
Train Accuracy: 0.5263315886772912      Train Loss: 1.5555219033154948
Running Evaluation: 100%|██████████| 15875/15875 [03:22<00:00, 78.59it/s]
Epoch - 1
Running Epoch : 100%|██████████| 2320/2320 [06:22<00:00,  6.07it/s]
Train Accuracy: 0.6882446890504196      Train Loss: 0.812912741001567
Running Evaluation: 100%|██████████| 15875/15875 [03:21<00:00, 78.60it/s]
Epoch - 2
Running Epoch : 100%|██████████| 2320/2320 [06:22<00:00,  6.07it/s]
Train Accuracy: 0.7790871305712338      Train Loss: 0.5150295378222804
Running Evaluation: 100%|██████████| 15875/15875 [03:21<00:00, 78.66it/s]
Word Error Rate-  [4.130015748031496, 3.6964409448818896, 3.6498267716535433]
```